Final Report for ONR

From: Dr. Michael Kuperstein, Symbus Technology Inc., 950 Winter St. #1900 MA 02154
Re: ONR contract #N00014-93-C-0239
entitled **Neural Networks that Create their Own Goals Using Growth Cycles**
Date: December 2, 1994

Over the past year our research on the generation and control of path planning has formed a new theory using neural networks that allows intelligent agents to automate sub-tasking in achieving desired goals. This theory called Growth Cycles provides functionality that will be crucial at the next stage of computing and communication systems with their exponentially increasing overhead. Distributed intelligent agents will begin to autonomously and adaptively maintain the huge and complex National Information Infrastructure. The specific research shows how an intelligent agent can learn to go from anywhere to anywhere around obstacles in novel and contingent environments. Although the research focuses on adaptive path planning, it can also be generalized and applied to adaptive and autonomous problem solving. The following paper details the entire effort for this contract and has been submitted for publication.

Artificial Neural Network Growth Cycles that Create Spatial Cognitive Maps

by Michael Kuperstein, Symbus Technology Inc., Waltham, MA 02154

An artificial neural network theory based on cycles of growth and performance is presented that allows an intelligent agent to learn how to go from anywhere to anywhere around obstacles in novel and contingent environments. A spatial cognitive map learns steps that go from a present state to a new state which either leads directly to a goal or in the past has led to the same goal. The map allows an agent to go from one place to multiple goals and from multiple places to the same goal. The map incrementally learns novel paths around new obstacles that appear after previous learning.

The brain can learn to navigate an organism so that it can go from anywhere to anywhere on terrains riddled with changing obstacles. It can create its own goals, define and plan its own task sequences and accomplish them in an uncertain world. Understanding how the brain does this and applying it to controlling intelligent agents involves mechanisms of autonomous task decomposition and adaptive path planning. Some previous work has cast this problem as optimal control (1) where policies map states to actions that achieve the agent's objective. The policies are determined by maximizing a functional of the payoffs received during some time period. If an exact world model and the payoff structure are available, dynamic programming based on computational procedures can be used to solve the optimal policy (2) or neural networks can be used to learn potential field gradients generated from obstacles or goals (3). However, most of the time, the world model and the payoff structure are either not known or too difficult to find. Control architectures based on reinforcement learning methods are increasingly being used for learning situation-action rules or reactions that can be used for decision making. These methods including temporal difference learning (4) and Q-learning (5) approximate dynamic programming techniques and can be used to estimate an optimal policy. They have led to some successful applications including learning to play backgammon (6) but they also have their problems. These methods either have not been scaled to more complex and/or multiple tasks or have memory requirements that grow much too fast to be practical. One reason is that with each new learned sequence step, updates of neural network weights are needed for many or all of the previous steps.

I present a neural network architecture and mechanism, called a growth cycle network, in which an intelligent agent learns behavioral sequences from cycles of incremental learning and performance using intrinsic motivational drives. The theory of growth cycles consolidates and extends ideas from psychological motivation and learning theories, developmental psychology, and artificial neural networks. The framework of the theory is inspired from observations of the balance and relations between stability and growth, between spatial and temporal events and among sensations (cues), behaviors, expectations, plans and drives.

In developmental psychology, Jean Piaget (7) analyzed the stages of child development and developed the view that: all knowledge is simultaneously accommodation to the object and assimilation to the subject. The progress of intelligence works in the dual directions of a perceived universe constantly becoming more external to the self and intellectual activity becoming progressively internalized. He talked about the temporal transformation of structures in the double sense of differentiation of substructures and their integration into totalities. A number of

psychologists believe that this type of integration is centered around intrinsic motivational drives. Deci and Ryan (8), who reviewed this literature, detailed how both intrinsic and extrinsic motivational drives might be combined for the integration of cognitive structures in human development.

Focusing closer on the development of single abilities in infants, Watson (9) studied the behavioral cycle he called "the game" in which infants' smiles and cooing were highly correlated with a successful solution to tasks in which stimuli were followed by some perceivable outcome. He observed that a neutral or positive stimuli first evoked exploratory behavior that led to an evaluation of outcomes. With increasing familiarity, clearly contingent stimuli caused positive emotional responses; non-contingent stimuli caused little response and ambiguous stimuli caused negative responses. In the study of similar cognitive growth cycles in infants, Elkind (10) hypothesized that the motive forces inherent in the formation of cognitive structures are largely dissipated once the cognitive structures are fully formed. As a consequence, completed cognitive structures developed during one growth cycle require either another growth cycle or other extrinsic motivational drives to exercise the cognitive structure.

Recently in neural networks and artificial life, a number of approaches to autonomous robots use drive reinforcement as a fundamental concept (11). However, these efforts have not yet led to solutions of practical problems. To focus and test the various concepts of the growth cycles theory, I have modeled a computer simulated world of an intelligent agent behaving in a novel terrain. To focus on sequence learning, I have chosen a model problem which does not involve issues of adaptive classification, attention and chunking of representations. Eventually, these other issues will need to be solved for practical mechanisms to be viable. The strongest assumptions in this model problem are made to minimize the involvement of these issues. However, the present solution to the problem is designed to allow extensions that can naturally incorporate these issues.

I assume that the intelligent agent starts out with a variety of sensations, movement abilities, drives and reinforcement conditions, and the ability to learn associations among its representations. These abilities are initially uncalibrated and uncoordinated relative to the environment. The dynamic environment is a cell world with landmarks, walls, obstacle(s) and goal(s) as shown in Figure 1. The agent can sense the angles and distances of what it sees all around it and represents them in radial topographic maps (12). Although it can not see past obstacles and walls, it can see landmarks in the distance above the obstacles and walls. The landmarks have no intended relationship to the goal. At the outset the agent has no world model, except that the agent is assumed to be able to classify whether something is a goal, obstacle or any individual landmark. It can not differentiate among obstacles or walls. The agent can move in single cell units in any one of 5 equally spaced directions from its current orientation in the range $\pm\pi/2$ radians, or move back a step or move back and face the opposite direction.

The agent's behavior in its world is determined by a process that cycles for each behavioral step. During each cycle, shown in figure 2, the agent is always assumed to be in some drive state. The process (numbered in figure 2) involves the following stages: 1 determine a drive state; 2 gate the expectations and plans; 3 sense a cue; 4 match or learn the cue to an expectation; 5-6 associate the expectation to a plan or explore; 7 act; 8 sense an outcome; and 9 match or learn an outcome to an expectation or 10 satisfy the active drive. In this problem, the drive is getting to a goal, but for other problems, drives may include avoiding danger, increasing autonomy, increasing relatedness and/or increasing self-benefit. The existence of drives assumes that evaluation mechanisms exist to determine when a cue satisfies a drive. In the process cycle, sensory inputs or cues and the drive state are first processed into possible plans. If there are no plans, exploration is

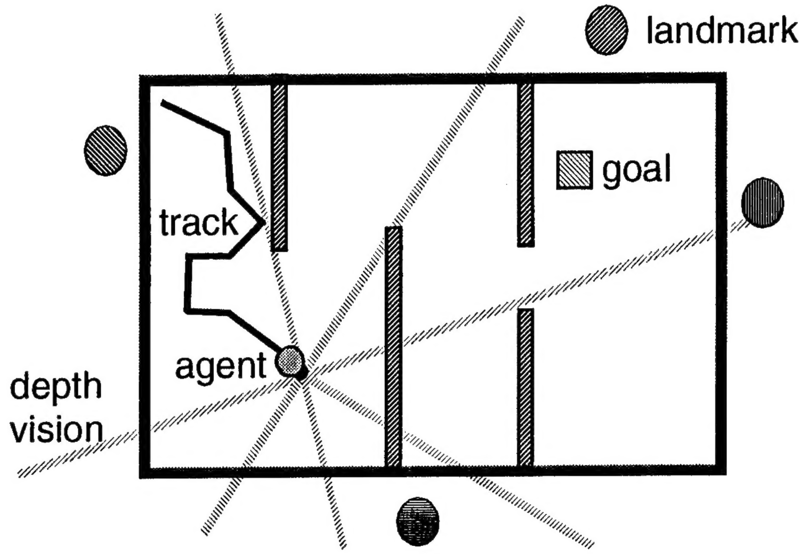

Figure 1. An intelligent agent in its novel

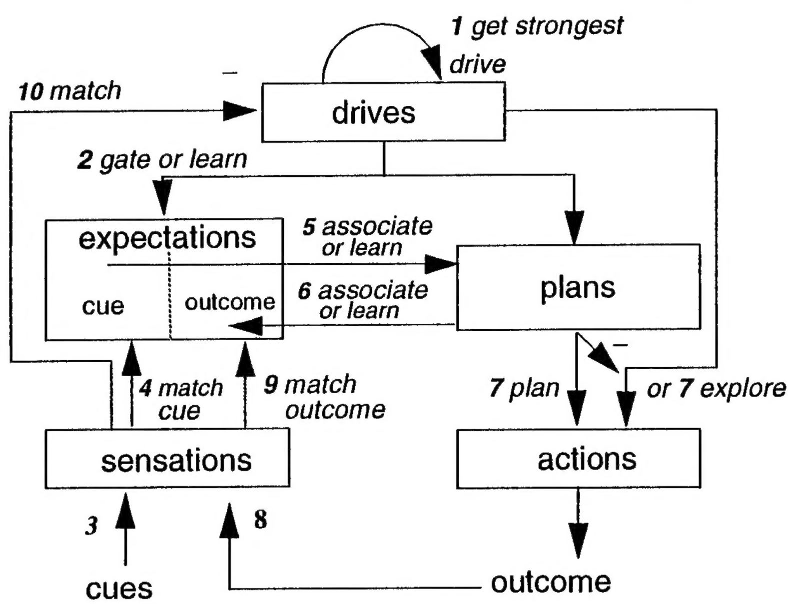

Figure 2. Schematic of the agent's representation architecture. The numbers are steps in a process that cycles for each movement step.

activated. The agent then behaves and changes some of the world cues which results in an outcome. The outcome is processed for learning new expectations, plans and associations. Then the next process cycle begins again.

To get around efficiently, there are three types of learning the agent must accomplish: avoiding obstacles, getting to a goal that it can sense, and of primary focus here, getting to a goal that it can not currently sense using sequential behavior. The adaptive control mechanisms to achieve these tasks, are arranged in a hierarchy with different priorities similar to the layered control system used by Brooks (13). The fixed priorities listed from high to low are: avoiding an obstacle, pursuing a sensed goal, pursuing a positive cue and performing a plan.

Avoiding obstacles adaptively can be accomplished using artificial neural networks (ANNs) (14). For this problem, when an obstacle is hit, the agent learns to associate the obstacle representation with the last move and then finds an escape move by exploration (15). If the agent can not find an escape move, it will step backward and move in the direction opposite to the direction in which it last moved. When the agent senses the same obstacle in the future, it will only move in a direction that does not match any move direction associated with the obstacle. Otherwise, a new move is randomly explored.

Reaching a sensed goal can also be accomplished with adaptive sensory-motor coordination using ANNs (16). Within the constraints of this system, the agent only needs to use a single joint control ANN to reach a goal that it can sense (17).

Even though every movement step to a new cell taken by an agent can be independent in this cell world, the agent moves across cells in two ways: The spatial map organizes the movements as either moving within a neighborhood in graded sensory-motor transitions or moving between neighborhoods in very sharp sensory-motor transitions. Moving from any point to any other point within a neighborhood without obstacles, can and has been achieved by ANNs that represent sensory-motor coordination and gradients (16). Also, selecting where to move among a number of neighborhoods separated by boundaries can also be achieved by popular ANNs that represent classifications of associations. Both types of representations will be required for sequence learning in this cell world.

In general, a behavioral sequence can be learned in three ways: 1. associating a string of actions which may be dynamically tuned by outcomes such as the walking sequence; 2. associating a string of cues with fixed cue-action responses such as list learning or 3. associating a string of adaptive cue-action pairs, which is the focus of this work. For path planning, associating a string of actions would be like walking through a maze with your eyes closed and hands tied behind your back. Such a sequence would be subject to drift from the accumulation of small movement errors and would fail to negotiate changes in the environment or varying states of an agent after learning. Associating a string of cues would be like trying to find your way in a forest where all the trees look the same. What you would need to learn is the order of the pattern of trees that you see in each step. This becomes difficult if there are many patterns that look the same in a sequence. These observations lead to three necessary conditions for adaptive sequence learning: 1. Cues need to be differentially perceived. 2. Both touch sensing and either telesensing or gradient sensing are required. Touch sensing is needed to experience hitting obstacles while telesensing or gradient sensing is needed to measure distance and direction between successive places in the world. 3. At least some contextual cues like landmarks need to be continually sensed, associated or assumed during all steps of a sequence. These contextual cues are required to provide a measure of environmental distance traveled in a planned action, even though they do not include the goal, nor do they represent any known prior relationship to the goal. A planned action may pass through vias that totally change

the sensory view. The contextual cues provide a sense of continuity that can connect two places with very different local cue sensations. The continuity of the contextual cues that occurs while moving between abruptly varying local cues allows a neural network to represent behavioral steps by local gradients as discussed below.

An adaptive behavioral sequence here is represented by a series of learned steps in the context of a drive and world cues. In a process called the growth cycle, each sequence step is learned as an association of three components: a learned beginning cue representation, called the cue expectation; a movement direction, called the plan; and an ending cue representation, called the outcome expectation. The drive chooses the sequence steps that will eventually lead to its satisfaction by classifying and gating the associations, similar to the modular connectionist architecture described by Jacobs and Jordan (18). An expectation or plan can be associated with other expectations or plans, where each association is gated by a different drive. In this way, multiple learned sequences that satisfy different drives can share the same cues in some of the sequence steps, without getting confused. A spatial cognitive map emerges from alternations between growth cycles where all of these representations and associations are learned and performance cycles where the learned associations unfold and new cues are perceived. As described below, each growth cycle bootstraps on previous growth and performance cycles.

The agent responds to cues depending on its behavioral state and on built-in, constant system policies. The states include exploring, pursuing a positive cue, escaping a negative cue (avoiding an obstacle), learning a sequence step and performing a sequence step. The default state is exploring. A growth cycle is a combination of pursuing a positive cue and learning a sequence step. The system policies determine which state the agent will be in, as well as determining the type, valence (positive or negative) and magnitude of learning reinforcers that result from outcome types. A positive outcome results when a transitory cue occurs for either of two events. The event, called the positive cue, can either be when the agent senses a goal or when the agent senses that it is close enough to a cue expectation. Being close to a cue expectation is reinforcing because the cue expectation represents the beginning of a step in a sequence that will lead to the goal. Being "close enough" to a cue expectation occurs when the similarity of the current cue sensation to a cue expectation (19) is above some threshold. A negative outcome results when a transitory cue occurs that stops the agent from getting to a goal. Learning each sequence step through a growth cycle starts when a reinforcing event or positive cue is perceived and finishes when the expected outcome is perceived or when a goal is reached.

When a positive cue is perceived and a growth cycle begins, the current cue sensation is first stored in memory to become a cue expectation for future use. The agent then pursues the positive cue until the agent either reaches it or cannot get any closer. If the agent cannot get to the positive cue, then whatever is temporarily stored in memory is forgotten. Along the way to reaching the positive cue, a new plan is learned. The plan is the accumulated, average movement direction, relative to a specific landmark perceived at the plan's onset. The plan allows the agent to traverse a straight distance between two places and usually involves multiple movement steps that are grouped together.

Reaching the positive cue is determined when the similarity between the expected positive cue and the actual current cue is above some threshold. When the agent gets there, a number of events happen. First, the accumulated plan and the specific landmark are stored in memory. Second, the current cue sensation is stored in memory to become an outcome expectation for future use. This may seem redundant, but note that the positive cue that the agent expects may not be the same as the actual cue when the agent gets close enough to the positive cue. Third, the links between the

cue expectation as input and the plan and outcome expectation as output, are learned. Fourth, all the links become gated by the current drive. The combination of all of these associations learned through a growth cycle define one sequence step in the spatial cognitive map.

To perform a sequence step, the agent first enters the performance state by matching the current sensation to the cue expectation above some threshold. Then the agent follows the associated planned direction until the current sensation matches the outcome expectation above another threshold. Then either the goal has been reached or the agent matches the next cue expectation to begin the next sequence step. Otherwise, it starts exploring. If the goal is reached, the drive becomes satisfied and a new drive state is determined by competing among the other drives.

To clarify the learning process I present an example of three learning growth cycles shown in Figures 3. Suppose an agent is exploring its space with some drive when it first senses the goal. That event starts the learning of the first cue expectation from the cue sensation at its current position. This expectation is stored temporarily in memory. As the agent pursues the goal, it accumulates an ongoing plan. When the agent reaches the goal, the growth cycle learns the plan and outcome expectation. Then it learns to associate the expectations and plan and learns to gate the association with the current drive.

Now suppose sometime later, the agent was again exploring its space with the same drive as before and it senses that it is close enough to the first cue expectation. That event starts the learning of the second cue expectation. The agent pursues the first cue expectation through a sensory-motor feedback loop. Just like sniffing your way closer to an odor source, successive sensation comparisons of cues to expectation eventually get the agent closer. If a successive comparison does not get closer during the pursuit, the agent backs up and tries a new direction. If no successive comparison gets closer or an obstacle gets in the way, the agent stops pursuing and starts exploring. During a successful pursuit, the agent accumulates its movement directions into a second ongoing plan. If the agent reaches the first cue expectation, the growth cycle learns a second outcome expectation and plan and then associates the two new expectations and plan all together. If the agent does not reach the first cue expectation, then the second cue expectation, which was stored in memory, is forgotten. The same cycle is repeated to learn the third step.

Over many growth cycles, a spatial cognitive map will emerge that represents forward behavioral sequences learned chronologically from the goal(s) backward. Alternatively, behavioral sequences can grow outward from the starting place if the goal that begins near the starting place is incrementally moved outward from with every successful learning run, like in animal training. In summary, the map learns steps that go from a present state to a new state which either leads directly to a goal or in the past has led to the same goal.

When a plan or pursuit is either achieved or abandoned, their step representation in the spatial map becomes inactive for some time, during which the representation can not be used to match against. This refractory period is necessary to avoid circular paths in the environment.

The loose coupling of sequence step representations allows an agent to reach a goal, starting anywhere in the sequence. However, if the match of a outcome expectation reliably leads to the match of the next cue expectation in the unfolding of a sequence, then it becomes more efficient to simply skip the matching of the next cue expectation and directly associate the current outcome expectation with the next plan. If the sequence is even more reliable, each plan can simply be associated with the next. This resembles the storage of an ordered sequence of movements. The more reliable and less contingent the steps become, the most efficient the sequence can get. However, for this increased efficiency, the middle cue associations will be lost and the agent will not be able to start in the middle of a sequence.

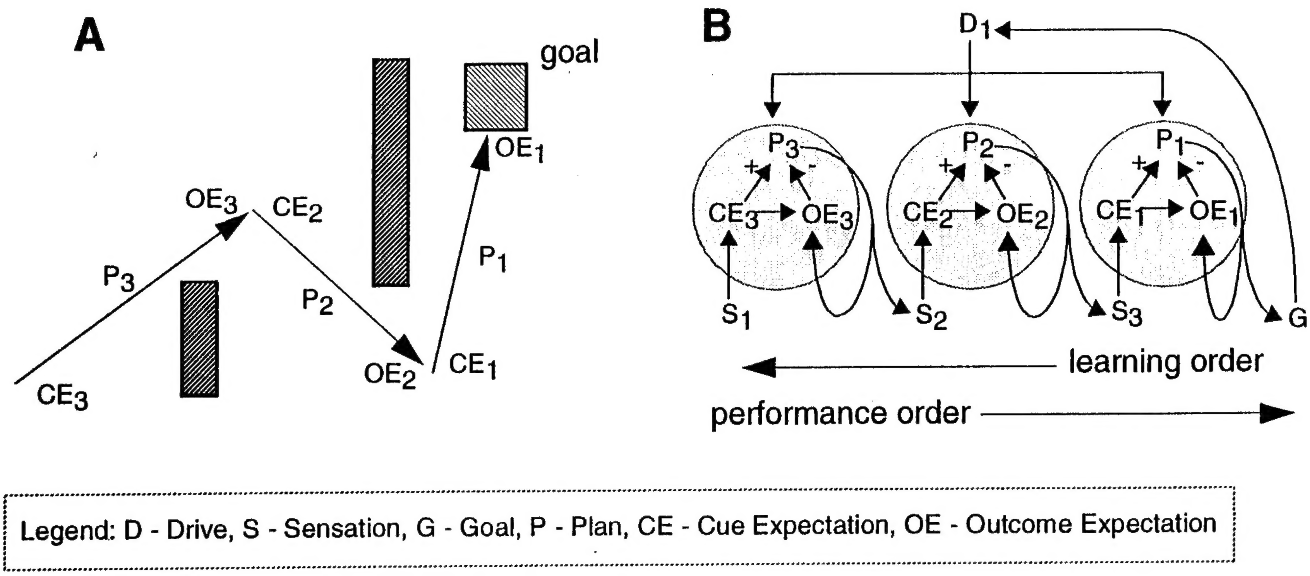

Figure 3. Two views of how three example sequence steps are learned and performed. (A) A geometric view: For each step in the sequence, the cue expectation, plan, outcome expectation and their associations are learned in order, in the context of the active motivational drive. This defines one growth cycle. One sequence step usually involves multiple movement steps that are grouped under one plan that traverses the distance between a cue expectation and an outcome expectation. The sequence steps are learned chronologically backwards from the goal and performed forward to the goal. (B) A representational view: Each circle is one sequence step learned by a growth cycle in the spatial map at one time. For each step, the sensation is matched against a cue expectation which initiates a plan. The plan is run until a sensation matches the outcome expectation. At that point the plan is stopped and the next sensation starts the next step. Once a step is done, it is temporarily inactive during a refractory period. The motivational drive enables each step until the drive is satisfied by the goal stimulus.

The choice of a specific ANN is not crucial to implementing reinforcement learning and a number of alternative ANNs could be used without much change in agent performance. A computer simulation of an intelligent agent in a maze tested some the properties of growth cycles. Figure 4A shows a graph of the number of movement steps the agent used to reach the goal across learning runs, when the agent starts from the same place. Figure 4B shows the path (of 42 movement steps, 9 sequence steps) that the agent used after its best learning occurred on the 12th run. The path itself is not optimal, nor is the spatial map designed to achieve optimal paths. It is designed more to achieve workable paths that are somewhat idiosyncratic to the choice of spatial map parameters such as the threshold of similarity between cue and expectation. Figure 4C shows the number of steps the agent used to reach the goal across learning runs, when the agent starts from random places. Because an agent can start a sequence in the middle, the set of possible sub-sequences in the context of one drive are all related and can be used as the environmental cues recall them. These results shows the ability for a spatial cognitive map to navigate an agent so that it can go from anywhere to a single goal. Going from anywhere to anywhere, including visiting goals in a specific order, can be achieved using multiple drives. In this initial formulation, the maximum number of sequences in one world context is the number of drives times all the combinations of the number of differentiable cue representations.

What happens if an obstacle appears in the path of a previously learned step? The agent will abandon the current plan and begin to explore until it picks up another possible cue expectation, which will allow the agent to learn to reach the goal through a different sequence. The abandoned plan will not be easily forgotten, in case the contingent obstacle is removed at a later time. However, sequence step representations will atrophy after long periods of non-use. Figure 4D shows the path (of 35 movement steps, 10 sequence steps) that the agent used when one of the obstacles was moved and blocked a previously learned path. This new path was learned after an additional 4 learning runs.

How does the spatial cognitive map scale with more complex problems? To extend the current formulation to multiple world contexts, each context would need to be independently classified and then each classification would adaptively gate all the sequences learned in that context. The memory requirements for this would grow quickly since every learned sequence step would need one context link, one drive link and two expectation-plan links. The efficiency of memory usage can be enormously improved by chunking and sharing the representations. In this model, drives already serve to chunk behavior sequences. By applying this model recursively in a hierarchy, each drive can represent a step in a higher level sequence and thus, higher level drives would be used to integrate and chunk sequences of lower level drives. Lower level sequences that satisfy similar goals across different world contexts could thus be shared. More work is required to specify the mechanism for when and how a new level of the hierarchy would be generated.

The growth cycle network, GCN, is most similar to a temporal difference network, TDN, (4), but there are some major differences. Whereas, TDN requires learning to propagate back through many or all the steps of a trial sequence, GCN learns to bootstrap one step at a time. Whereas TDN seeks the optimal minimal cost for sequences, GCN seeks workable sequences. These differences make a GCN learn faster than a TDN. Because of the interactions between cues, expectations and plans a GCN can be used distinguish between three types of contingencies and sources of error: cues, plans and/or outcomes; while TDN only account for changes in cues. This makes GCN more flexible in dealing with variability in the world and/or in an agent. Moreover, a TDN treats each movement step independently, while a GCN chunks a number of movement steps with each learned plan. This leads to great savings in memory requirements.

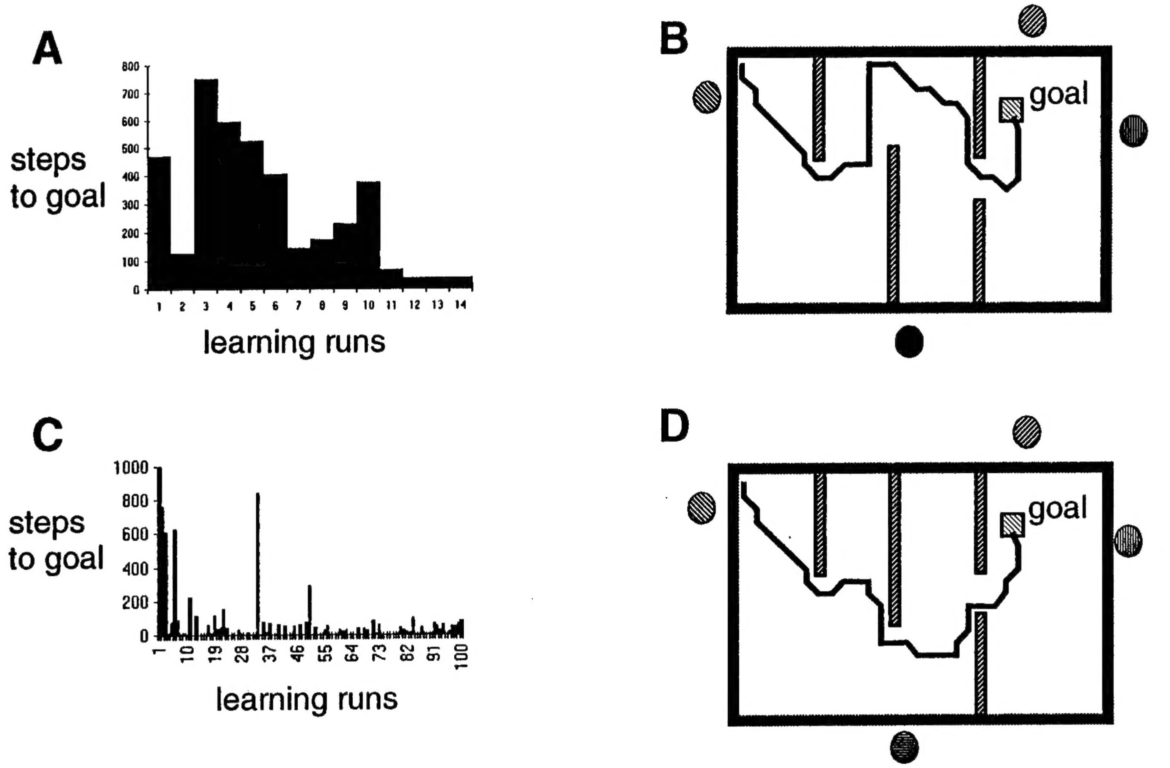

Figure 4. (A) A graph of improving performance across learning runs when the agent always starts from the same place. (B) The agent's path to the goal (42 movement steps, 9 sequence steps) after 12 learning runs. (C) A graph of improving performance across learning runs when the agent starts from random places. (D) Here, one of the obstacles has been moved relative to the cell world shown in C, after the path in C has been learned. After 4 more learning runs, a new path to the goal (35 movement steps, 10 sequence steps) is learned. Both learned paths can now be used depending on where that obstacle is placed.

The ability for an intelligent agent to build a cognitive spatial map can be applied not only to adaptive navigation, but may also be a key to adaptive problem solving. In navigation, an agent senses different cues as it moves through its world while in general problem solving, an agent senses different cues as it transacts with the world. Each transaction transforms which cues are perceived. To extend adaptive path planning to problem solving using growth cycles would require an analogy to the similarity measure used here between both cues and expectations. The similarity measure used here is related to the physical distance-to-goal, while a general similarity measure between the classifications of cues used for a problem solving sequence would be related to the progress-to-solution.

The distributed and intrinsic representations of drives, cues, plans and expectations has been designed to allow extensions of the growth cycle theory to include hierarchical classification representations and attentional mechanisms. The eventual goal of extending this line of work is to provide architectures and mechanisms by which intelligent organisms or computing agents can progressively predict and control their changing world and gain self-benefit by internalizing their transactions with the world and other agents.

Acknowledgements

This work was made possible by a contract from the Office of Naval Research #N00014-93-C-0239. I thank Dr. James Kottas for his comments in preparing the manuscript.

REFERENCES AND NOTES

1. A.G. Barto, R.S. Sutton and C.J.C.H. Watkins in *Learning and Computational Neuroscience,* M.Gabriel and J.W. Moore, Eds.(MIT Press., Cambridge, MA 1990).
2. Bellman, R.E. *Dynamic Programming* (Princeton University Press, Princeton, NJ,1957).
3. Thrun, S.B, & K. Moller in *Advances in Neural Information Processing Systems 4*, J.E. Moody, S.J. Hanson, and R.P Lippmann, Eds. (Morgan Kaufmann Publishers, San Mateo, CA., 1992).
4. R.S. Sutton, *Machine Learning,* **3**, 9 (1988).
5. C.J.C.H.Watkins, thesis, Cambridge University, Cambridge, U.K.(1989).
6. G. Tesauro, *IJCNN Proceedings III*, 33 (1990).
7. J. Piaget, *The Origins of Intelligence in Children*, translated by M. Cook, (International University Press, New York,1952); J. Piaget, *The Construction of Reality in the Child,* Translated by M. Cook, (Ballantine Books, New York, NY,1954); J. Piaget, *Biology and Knowledge* (University of Chicago Press, Chicago, IL,1971).
8. E. L. Deci and R. M. Ryan, *Intrinsic Motivation and Self-Determination in Human Behavior,* (Plenum Press, New York, NY, 1985).
9. J.S.Watson, *Merrill-Palmer Quarterly of Behav. & Dev.,* **18**,323(1973).
10. D. Elkind, in *Nebraska Symposium on Motivation*, J. K. Cole, Ed.(University of Nebraska Press, Lincoln, NB,1971).
11. J.G. Taylor, *Neural Networks,* **7:6/7**,1181(1994).; J.-A. Meyer and A. Guillot *Proceed. 1st Conf. From Animals to Animats*, 1(1991).

12. The maze has a grid size of 20 by 30 units. Visual representations are generated with the assumption that depth can be perceived. The representations are composed of radial surface maps where the radial angle is the angle of what is sensed relative to the agent's orientation, in the range $-\pi$ to π, and the radial distance is proportional to the distance of what is sensed from

0 to the maximal sense range that depends on the specific sense type. The near sense range was set to 3 grid units. The goal sense range was set to 12 grid units and the landmark sense range was set to 38 grid units. Each map represents a perceived input point, $I_{\rho,\theta}$ as a monopolar amplitude distribution, $R_{i,j}$ whose peak is centered at the perceived distance, ρ, and radial angle, θ, and that spreads out to some fixed distance depending on a constant, κ, such as in the cone shape:.

$$R_{ij} = 1 - \kappa\sqrt{(i-\rho)^2 + (j-\theta)^2} \qquad 0 \le R_{ij} \le 1 \qquad (1)$$

The form of the gradient distribution can be nonlinear and asymmetric without much effect on the results.

13. R. A. Brooks, *IEEE Journal of Robotics and Automation*, RA-2, April, 14(1986).
14. T.J. Prescott and J.E.W. Mayhew in *Advances in Neural Information Processing Systems 4*, J.E. Moody, S.J. Hanson, and R.P Lippmann, Eds.(Morgan Kaufmann Publishers, San Mateo, CA, 1992) p523.
15. The obstacle avoidance scheme used here, was designed to be adequate but not necessarily robust, since the focus of this work is more on learning sequences. It was implemented by an ANN that learns to associate visual representations of nearby obstacles with representations of move consequences. The form of the association is a simple link between the representations. There are 5 possible moves in equal angles relative to the agent's current orientation, in the range $-\pi/2$ to $\pi/2$. Single trial learning occurs whenever the agent hits an obstacle. First, the current visual representation is matched against all previously learned obstacle expectations to find the best match above a similarity threshold (19). If a close enough association has been learned, it is updated. Otherwise, a new association is created, all of its move values are initially set to 0 and then updated. For each move consequence, a value of 1 represents that the move will hit an obstacle and a value of 0 represents that it will not. The move consequence that caused the hit is updated during learning by setting its value to 1.
16. M. Kuperstein, *Proc. IEEE Internat. Conf. Automat. Robotics*, March, Raleigh, NC, 1595 (1987); M. Kuperstein, *Science*, **239**,1308(1988); M. Kuperstein, *Neural Networks*, **4**, 131 (1991).
17. Goal pursuit for this single joint agent can be implemented by any one of a number of ANNs. The ANN learns to adapt a gain element that transforms the sensed angle of the center of visual contrast to a rotation angle that orients the agent to point to the visual center. An example ANN has a single layer, a linear transform function and a single weight value.
18. R.A. Jacobs and M.I. Jordan, *IEEE Trans. Systems Man Cyber.*, **23:2**, 337 (1993).
19. The measure of similarity, S, between representations of a cue, R_C, and representations of a learned expectation (of either cue or outcome type), R_E, uses a normalized dot-product function across all corresponding values, i,j:

$$S = \frac{\sum_{i,j} (R_{C(i,j)} \cdot R_{E(i,j)})}{\sqrt{\sum_{i,j} (R_{C(i,j)})^2} \cdot \sqrt{\sum_{i,j} (R_{E(i,j)})^2}} \qquad (2)$$

The sensed cues at one location vary with the orientation of the agent. Since the closeness of a

sensed cue to an expectation should not depend on which direction the agent is facing, the complete set of 8 possible sensory representations, one for each possible orientation, make up a cue representation. For more complex and practical models with much higher directional resolution, does this mean that a cue representation will need thousands of possible sensory representations? No, additional attentional and hierarchical mechanisms, not discussed here, will efficiently represent the cue as invariant to the focus direction and will minimize memory usage.